

# **PREDICT DIABETES USING MACHINE LEARNING MODELS**

A Capstone Phase project report submitted  
in partial fulfillment of requirement for the award of degree

## **BACHELOR OF TECHNOLOGY**

in

## **COMPUTER SCIENCE & ENGINEERING**

by

18K41A05A7	Singareddy Akshitha
18K41A05B0	Singirikonda Niharika
18K41A05B6	Veeramneni Pravallika
19K45A0506	Kandula Akhil
17K41A0576	Namindla Tejaswini

Under the guidance of

**K.Sudheer Kumar**

Assistant Professor, Department of CSE

**Submitted to**



**SR**  
**Engineering**  
**College**  
Innovation . Creativity . Entrepreneurship

# **SR ENGINEERING COLLEGE**

Ananthasagar, Warangal.



## **CERTIFICATE**

This is to certify that this project entitled “**PREDICT DIABETES USING MACHINE LEARNING MODELS**” is the bonafied work carried out by **Singareddy Akshitha(18K41A05A7), Singirikonda Niharika(18K41A05B0), Veeramneni Pravallika (18K41A05B6), Kandula Akhil(19K41A0506), Namidla Tejaswini(17K41A0576)** as a Capstone Phase-II project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING** during the academic year 2021-2022 under our guidance and Supervision.

**K.Sudheer Kumar**

Asst. Prof.(CSE),  
S R Engineering College,  
Ananthasagar, Warangal.

**Dr. M.Sheshikala**

Assoc.Prof.& HOD(CSE),  
S R Engineering College,  
Ananthasagar, Warangal.

**External Examiner**

## ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide **Mr.K.Sudheer Kumar, Asst.Prof.** as well as Head of the CSE Department **Dr. M.Sheshikala, Associate Professor** for guiding us from the beginning through the end of the Capstone Phase-II project with their intellectual advice and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to project coordinators **Mr. Sallauddin Md, Asst. Prof, Y.Chanti Asst. Prof.** for their encouragement and support.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

## **ABSTRACT**

Diabetes is considered as one of the deadliest and chronic diseases which causes an increase in blood sugar. Now-a-days one in ten adults are suffering from diabetes. Many complications occur if diabetes remains untreated and unidentified. The risk factor and severity of diabetes can be reduced significantly if precise early prediction is possible. Machine Learning plays a significant role in predicting healthcare Problems. The Aim of the project is to develop a system which can perform early prediction of diabetes for a patient with higher accuracy by combining the results of different machine learning techniques.

# CONTENTS

<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>LIST OF ACRONYMS</b>	<b>viii</b>

<b>Chapter No</b>	<b>Title</b>	<b>Page No</b>
1.	<b>INTRODUCTION</b>	<b>01</b>
	1.1 OVERVIEW	01-04
	1.2 EXISTING METHODS	04
	1.3 OBJECTIVE	04
	1.4 LITERATURE SURVEY	05-07
2.	<b>HARDWARE AND SOFTWARE TOOLS</b>	<b>08-10</b>
3.	<b>PROJECT IMPLEMENTATION</b>	<b>11</b>
	3.1 PROPOSED METHODOLOGY	11
	3.1.1 DATA COLLECTION	11-12
	3.1.2 DATA VISUALIZATION	12-13
	3.1.3 DATA PROCESSING	13-14
	3.1.4 APPLY MACHINE LEARNING TECHNIQUES	14-19
	3.1.5 MODEL BUILDING	19-20
4.	<b>SIMULATION RESULTS &amp; ANALYSIS</b>	<b>21</b>
	4.1 RESULTS	21-28
5.	<b>CONCLUSION</b>	<b>29</b>
	5.1 CONCLUSION	29
	5.2 FUTURE SCOPE	29
	REFERENCES	30-31

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>DESCRIPTION</b>	<b>PAGE NO</b>
<b>1.1</b>	Symptoms of Diabetes	<b>02</b>
<b>3.1</b>	Overview of the process	<b>11</b>
<b>3.1.2(a)</b>	Count plots graph of Data	<b>13</b>
<b>3.1.2(b)</b>	Histogram of Dataset	<b>13</b>
<b>3.1.4(a)</b>	Random Forest	<b>15</b>
<b>3.1.4(b)</b>	Logistic Regression	<b>16</b>
<b>3.1.4(c)</b>	SVM	<b>17</b>
<b>3.1.4(d)</b>	KNN	<b>18</b>
<b>3.1.4(e)</b>	Decision Tree	<b>18</b>
<b>3.1.4(f)</b>	Naive Bayes	<b>19</b>
<b>3.1.5</b>	Web Page	<b>20</b>
<b>4.1(a)</b>	Result1	<b>28</b>
<b>4.1(b)</b>	Result2	<b>29</b>

## LIST OF TABLES

TABLE NO	DESCRIPTION	PAGE NO
1.4	Comparison of Algorithms	07
3.1	Data Set	12
4	Results of Algorithms	21

## LIST OF ACRONYMS

ML	MACHINE LEARNING
SVM	SUPPORT VECTOR MACHINE
KNN	K-NEAREST NEIGHBOUR



# 1. INTRODUCTION

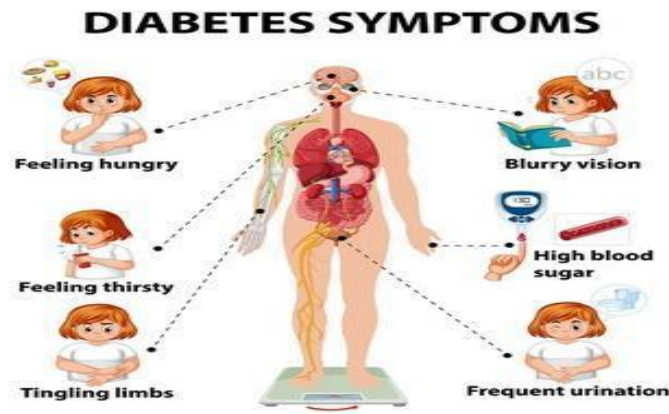
## 1.1 OVERVIEW

Diabetes is one of the deadliest and chronic diseases in the world which is rapidly increasing. It is not only a disease but also a creator of many diseases. Diabetes occurs when the human pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces, which leads to an increase in the blood glucose levels. Generally a person is considered to be suffering from diabetes, when blood sugar levels are above normal.

Basically there are two types where diabetes type 1 spans 5 to 10% of all diabetes cases. This type of diabetes appears most often during childhood or adolescence and is characterized by the partial functioning of pancreas. At the beginning, type 1 diabetes does not develop any symptoms, as the pancreas remains partially functional. The disease only becomes apparent when 80-90% of pancreatic insulin-producing cells are already destroyed. It mainly affects the organs like eyes, tiny blood vessels, kidneys, heart and nerves etc.

The other is type 2 diabetes, which presents 90% of all diabetes cases. In this condition pancreatic cells fail to produce sufficient amount of insulin, later as the disease progresses lack of insulin develops and the cells become insulin resistance. It is a mild form of diabetes but it may produce high risk of health complications affecting the small blood vessels which serve the organs such as kidneys, eyes, nerves and heart. Persons may have high risk with type 2 diabetes due to being overweight and less or no exercise. Type 2 diabetes is not curable however can be controlled with regular exercise, normal maintenance of weight, healthy diet and avoiding tobacco usage. Type 2 diabetes most commonly occurs in middle aged people and elderly people.

Furthermore, Gestational diabetes is an exceptional type of diabetes that tends to occur in pregnant women due to high sugar levels as the pancreas doesn't produce sufficient amounts of insulin. Ideally, 2 -10% of pregnant women are affected with diabetes. After the delivery of the baby she may not have diabetes or may lead to type 2 diabetes. It generally resolves after the birth of a baby.



**Figure 1.1 Symptoms of Diabetes**

In medicine, doctors and current research confirm that if the disease is discovered at an early stage, the chances of recovery will be greater. With the continuous advancement of technology, machine learning and deep learning techniques have become very useful in early prediction and disease analysis.

Our main objective is to predict diabetes at the prior stage. We used random forest and logistic Regression, Naive Bayes, SVM machine learning algorithms. Performance is compared using different parameters to achieve accuracy.

### **MACHINE LEARNING:**

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

### **What are the different types of machine learning?**

Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions.

There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

- **Supervised learning:** In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

- **Unsupervised learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
- **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
- **Reinforcement learning:** Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

### How does supervised machine learning work?

Supervised machine learning requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

- **Binary classification:** Dividing data into two categories.
- **Multi-class classification:** Choosing between more than two types of answers.
- **Regression modeling:** Predicting continuous values.
- **Ensembling:** Combining the predictions of multiple machine learning models to produce an accurate prediction.

### How does unsupervised machine learning work?

Unsupervised machine learning algorithms do not require data to be labeled. They sift through unlabeled data to look for patterns that can be used to group data points into subsets. Most types of deep learning, including neural networks, are unsupervised algorithms. Unsupervised learning algorithms are good for the following tasks:

- **Clustering:** Splitting the dataset into groups based on similarity.
- **Anomaly detection:** Identifying unusual data points in a data set.

- **Association mining:** Identifying sets of items in a data set that frequently occur together.

### **How does reinforcement learning work?**

Reinforcement learning works by programming an algorithm with a distinct goal and a prescribed set of rules for accomplishing that goal. Data scientists also program the algorithm to seek positive rewards -- which it receives when it performs an action that is beneficial toward the ultimate goal - and avoid punishments -- which it receives when it performs an action that gets it farther away from its ultimate goal. Reinforcement learning is often used in areas such as:

- **Robotics:** Robots can learn to perform tasks the physical world using this technique.
- **Video gameplay:** Reinforcement learning has been used to teach bots to play a number of video games.

## **1.2 EXISTING METHODS**

The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day to get their reports.

Diabetes prediction can be done by using different Data Mining Techniques, Machine Learning Algorithms and Deep Learning methods.

## **1.3 OBJECTIVE**

### **1.3.1 Motivation and Scope**

Diabetes is a growing health issue because of our inactive lifestyle. If diagnosed early and with appropriate treatment, side effects can be prevented. To facilitate early detection, technology can be used reliably and effectively. By using machine learning we can build a predictive model that can predict whether a patient has diabetes or not.

### **1.3.2 Problem Statement**

As a very common and rapidly growing disease, diabetes affects a large number of people from all ages each year reducing their life expectancy. Having a higher level of touch, increases the value of the initial diagnosis. Diabetes brings other complex problems such as cardiovascular disease, kidney failure, stroke, damage to vital organs etc. Early diagnosis of diabetes reduces the risk of developing an incurable and serious condition. Diagnosis and analysis of risk factors for various spinal

attributes helps to identify the prevalence of diabetes in medical diagnosis. The prevalence of diabetes in the early stages reduces the risk of future complications.

## **1.4 LITERATURE SURVEY**

Perveen, Sajida; Shahbaz, Muhammad [1] classification is done using diabetes risk factors followed by the data mining techniques (adaboost and bagging) and standalone J48 (c4.5) decision tree. The Classification is done across three different ordinal adult groups (18-35, 35-55, older than 55) in Canadian Primary Care Sentinel Surveillance network. The parameters they have taken are age, sex, systolic blood, diastolic blood pressure, high density lipoprotein (HDL) triglycerides (TRG), body mass index (BMI), and fasting blood glucose (FBG). Out of 667,907 patients, 40,042 patients were diagnosed as diabetic, which constitutes about 6% of the total patients. They observed that the older age group had more chances to get diabetic and concluded that age plays a major role in predicting diabetes. Furthermore, the results of adaboost are better compared to other techniques

Deepti[2] proposed three machine learning classification algorithms namely Decision Tree, SVM and Naive Bayes are used to detect diabetes at an early stage. The performances of all the three algorithms are evaluated on various measures like Precision, Accuracy, F-Measure. Accuracy is measured over correctly and incorrectly classified instances. Results obtained show Naive Bayes outperforms with the highest accuracy of 76.30% compared to other algorithms. These results are verified using Receiver Operating Characteristic (ROC) curves in a proper and systematic manner

Amani Yahyaoui [3] proposed a medical Decision support system for diabetes prediction based on Machine Learning (ML) techniques and also compared conventional machine learning with deep learning methods. For conventional machine learning methods, they considered the classifiers: Support Vector Machine (SVM) and also the Random Forest (RF), for Deep Learning (DL) made use of the Convolutional Neural Network (CNN) to predict and identify the diabetes patients. They took the Dataset of 768 instances out of which 500 instances belong to non diabetic class and remaining 268 instances were diabetic patients. The experimental results showed that RF was more effective for

diabetes prediction compared to deep learning and SVM methods. RF produced a complete diabetic prediction of 83.67%. SVM predictive accuracy reached 65.38% while DL method produced 76.81% in the database

Pradeep Kandhasamy, S. Balamurali [4] They took 4 classifiers Decision Tree J48, KNN Classifier, Random Forest, Support Vector Machine. The performances of the algorithms have been measured

in both the cases i.e dataset with noisy data (before pre-processing) and dataset set without noisy data (after pre-processing) and compared in terms of Accuracy, Sensitivity, and Specificity. They observed that after removing noisy data and undergone preprocessing technique it provided more accurate results and noticed that KNN( $k=1$ ) and Random Forest performance is more accurate than the other classifiers.

Kannadasan [5] proposed a Deep learning technique for diabetes data classification by Deep Neural Network (DNN) method using stacked autoencoders. Fine tuning of the network is done using backpropagation in supervised fashion with the training dataset. The proposed system has shown good classification accuracy of 86.26%.

Suresh Kumar [6] diabetes dataset is taken and experimented with Random Forest (RF), SVM, k-NN, CART and LDA algorithms. They selected these algorithms to pick out up semi-random algorithms using diversity of representation and mode of learning style. As a result it showed that RF algorithm is predicting the outcomes correctly.

Asma A. AlJarullah [7] presented a method to diagnose Type-2 diabetes by using Weka's J48 decision tree classifier. The dataset used is "The Pima Indians Diabetes Data Set". decision tree method was used to predict diabetes. Weka Software was used throughout all the phases of the study. The accuracy of the model was 78.176%.

B.M. Patil; R.C. Joshi; Durga Toshniwal [8] They proposed a Hybrid Prediction Model (HPM) model which uses Simple K-means clustering algorithm. C4.5 algorithm is used to build the final classifier model by using the k-fold cross-validation method. and observed that obtained results of HPM to predict diagnosed patients likelihood to get diabetic in next 5 years to a group that doesn't get diabetic showed a 92.38% accurate

Changsheng Zhu [9] proposed a data mining based model for early diagnosis and prediction of diabetes using the Pima Indians Diabetes dataset. The novel model consists of using PCA (principal Component Analysis) for dimensionality reduction, k-means for clustering, and logistic regression for classification. As a result they observed that performance of the improved logistic regression model was predicted at an accuracy rate of 89%.

Swathi Lakshmi [10] has proposed a method for diagnosing a patient's risk of diabetes. The patient's level of risk was determined using ontology and machine learning methods. Ontology has symptoms of the disease, its causes and treatment. In machine learning, naive bayes algorithm and decision tree are used to identify the disease type and Stage.

<b>REFERENCE NO</b>	<b>ALGORITHM USED</b>	<b>ALGORITHM ACCURACY(RESULT)</b>
[1]	adaboost , bagging and standalone J48 (c4.5) decision tree.	adaboost is better compared to other techniques
[2]	Decision Tree, SVM and Naive Bayes	Naive Bayes outperforms with the highest accuracy of 76.30%
[3]	Support Vector Machine (SVM) , Random Forest (RF), Convolutional Neural Network (CNN)	RF-83.67%  SVM-65.38%  DEEP LEARNING(CNN)-76.81%
[4]	Decision Tree J48, KNN Classifier, Random Forest, Support Vector Machine	Random Forest performance is more accurate than the other classifiers.
[5]	Weka's J48 decision tree classifier.	The accuracy of the model was 78.176%.
[6]	PCA (principal Component Analysis), k-means, and logistic regression.	logistic regression model predicted at accuracy rate of 89%.

**Table 1.4 Comparison of Algorithms**

## 2. HARDWARE AND SOFTWARE TOOLS

### 2.1 HARDWARE TOOLS

- System
- Hard Disk
- Ram-4 GB
- Processor

### 2.2 SOFTWARE TOOLS

- Operating System-Windows 10
- Jupyter Notebook
- Python IDLE(3.9 64 bit)
- Numpy
- Pandas
- Seaborn
- Matplotlib
- Flask
- Scikit Learn

#### **Jupyter Notebook:**

Jupyter notebook is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning. A modular design invites extensions to expand and enrich functionality.

#### **Numpy:**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions



- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

### **Installation:**

- **Mac** and **Linux** users can install NumPy via pip command:

```
pip install numpy
```

### **Pandas:**

- Pandas is a Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.
- Pandas allows us to analyze big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.
- Relevant data is very important in data science.

### **Seaborn:**

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

### **Matplotlib:**

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.

- Customize visual style and layout.

**Scikit Learn:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

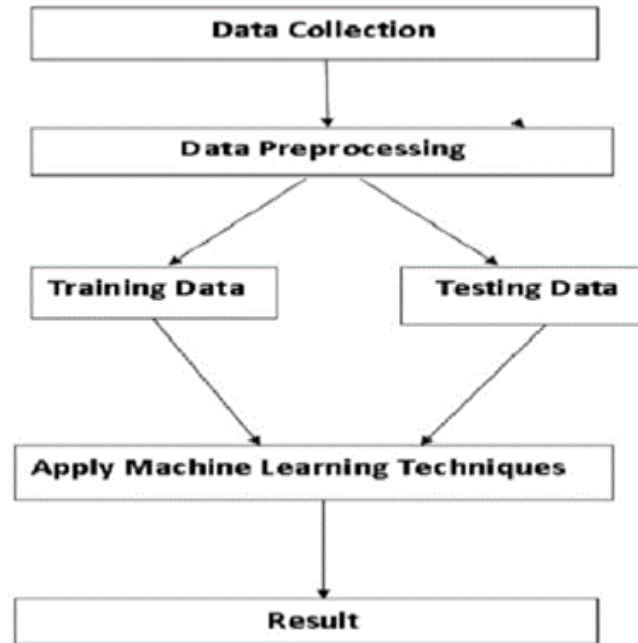
**Flask:**

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.

### 3. PROJECT IMPLEMENTATION

#### 3.1 PROPOSED METHODOLOGY

The below fig, shows an architecture diagram for diabetes prediction model.



**Figure 3.1 Overview of the process**

##### 3.1.1 Data Collection

The proposed diabetes classification and prediction algorithm is evaluated on a publicly available PIMA Indian Diabetes dataset. The primary objective of using this dataset is to build an intelligent model that can predict whether a person has diabetes or not, using some measurements included in the dataset. There are eight medical predictor variables and one target variable in the dataset. Diabetes classification and prediction are a binary classification problem.

The dataset consists of 2000 records of different healthy and diabetic female patients of age greater than twenty-one. The target variable outcome contains only two values, 0 and 1. The primary objective of using this dataset was to predict diabetes diagnostically. Whether a user has a chance of diabetes in the coming four years in women belongs to PIMA Indian. The dataset has a total of eight variables: glucose tolerance, no. of pregnancies, body mass index, blood pressure, age, insulin, skin thickness and Diabetes Pedigree Function. All eight attributes shown in Table are used for the training dataset in the classification model in this work.

S No.	Attributes
1	Pregnancies
2	Glucose Level
3	Blood Pressure
4	Skin thickness
5	Insulin
6	BMI (Body Mass Index)
7	Diabetes Pedigree Function
8	Age

**Table 3.1.1 Dataset**

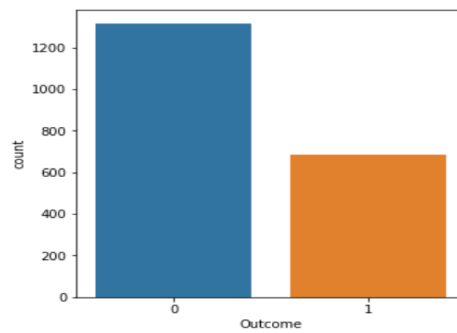
The 9<sup>th</sup> attribute is outcome. The outcome is the feature we are going to predict, 0 means No diabetes, 1 means diabetes.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

### 3.1.2 Data Visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

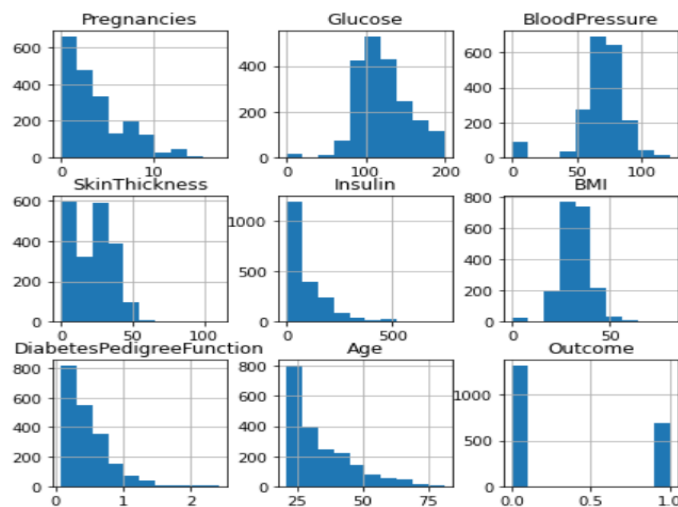
We made a model to predict diabetes however the dataset was slightly imbalanced having around 1316 classes labelled as 0 means negative means no diabetes and 684 labelled as 1 means positive means diabetic.



**Figure 3.1.2(a) Count plots of Data**

### **Histogram:**

A frequency distribution shows how often each different value in a set of data occurs. A histogram is the most commonly used graph to show frequency distributions. It looks very much like a bar chart, but there are important differences between them. This helpful data collection and analysis tool



**Figure 3.1.2(b) Histogram of Data**

### **3.1.3 Data Preprocessing**

Data preprocessing is the most important process. Mostly healthcare related data contains missing value and other impurities that can cause effectiveness of data. To improve quality and effectiveness obtained after the mining process, Data preprocessing is done. To use Machine Learning Techniques on the dataset effectively this process is essential for accurate result and successful prediction.

**i) Missing Values removal-** Remove all the instances that have zero (0) and null as worth. Having zero and null as worth is not possible. Therefore this instance is replaced with the mean value.

**ii) Splitting of data-** After cleaning the data, data is normalized in training and testing the model. We have split the data into training 80% and testing 20%. Then we train algorithms on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data.

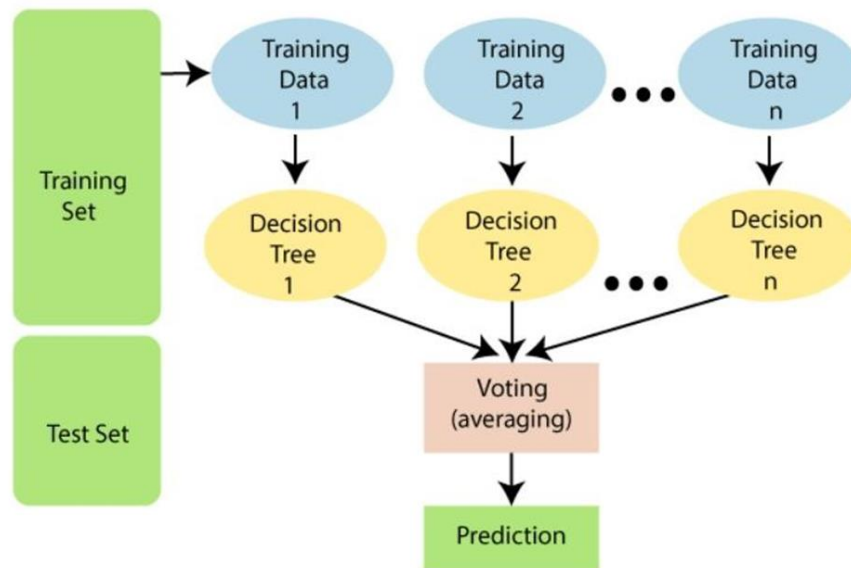
### **3.1.4 Apply Machine Learning Techniques**

When data has been ready we apply Machine Learning Technique. We use different classification and ensemble techniques to predict diabetes. The methods applied on Pima Indians diabetes dataset. Main objective is to apply Machine Learning Techniques to analyze the performance of these methods and find accuracy of them, and also be able to figure out the responsible/important features which play a major role in prediction. The Techniques we are using are Random Forest, Logistic Regression, svm, naive Bayes, knn, Decision Tree, The Techniques are follows-

#### **i) Random Forest (RF)**

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems. As its name implies, it is a collection of models that operate as an ensemble. The critical idea behind RF is the wisdom of the crowd, each model predicts a result, and in the end, the majority wins. It has been used in the literature for diabetic prediction and was found to be effective.

We are applying a random forest algorithm to classify the diabetes dataset. Random Forest is a popular machine learning algorithm that belongs to supervised learning. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The below diagram explains the working of the Random Forest algorithm



**Figure 3.1.4(a) Random Forest**

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps :

Step-1: Select random n data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Some advantages of Random Forest Algorithm are-

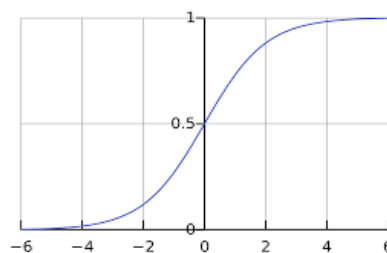
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

## ii) Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. Which refer to a case to classify a patient that is positive or negative for diabetes. It is appropriate to use logistic regression when the dependent variable is binary, as we have to classify an individual in either type 1 or type 2 diabetes. Besides, it is used for predictive analysis and explains the relationship between a dependent variable and one or many independent variables.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.



**Figure 3.1.4(b) Logistic Regression**

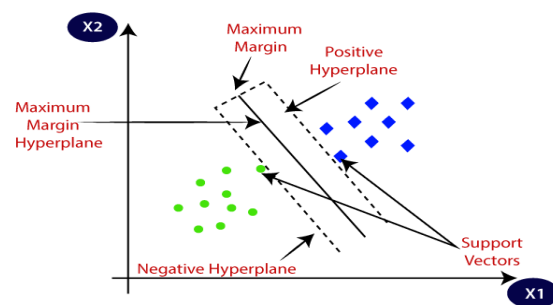
## iii) Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.



The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane



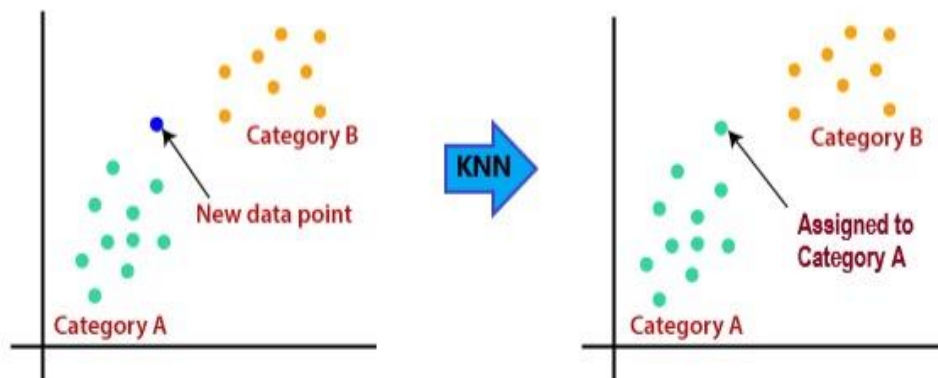
**Figure 3.1.4(c) Support vector Machine**

Here in this method, we complete the classification by defining the hyperplane amongst the characteristics. Each data item is interpreted as a point, and they are plot versus n-dimensional space such that the value of each feature being the value of a particular coordinate. Here the use of SVM is to depreciate the error rate and misclassification by recognising hyperplanes with high margins from the data point. SVM are instrumental in case of high dimensional space. Individual attributes have coordinates which are formally comprehended as support vectors. To accomplish the data transformation from the lower-dimensional input space towards higher-dimensional input space, we use kernel function, which encourages to tackle such complex change.

#### **iv) KNN (k-nearest neighbour)**

It is a simple, versatile and straightforward to implement supervised learning algorithm. It works on the ideology that similar observations are close to each other. It captures the concept of similarity by calculating the separation within two points on a graph. The 'k' in the algorithm is a numerical value that tells how many data points to consider for taking a vote. To classify a new point, we encircle the point with K number of datapoints and assign it to the group with the maximum number of points within the circle. The ideal way to identify the value of K is by trying out a few values of k before settling on one, which reduces the error and concurrently maintains

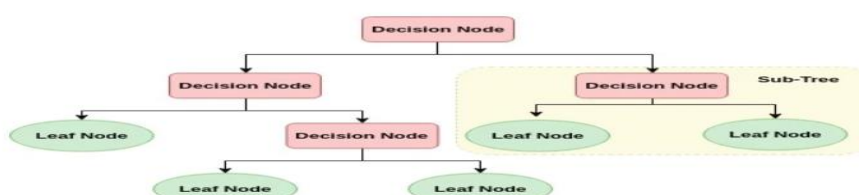
the accuracy of the prediction. Low values can be noisy and subject to outliers. Large values of  $K$  smooth over things but  $K$  should not be so large that other categories shall always outvote a category with a few examples.



**Figure 3.1.4(d) KNN**

#### v)Decision Tree

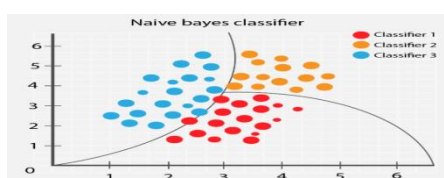
Decision tree is a basic classification and regression method. Decision tree model has a tree structure, which can describe the process of classification instances based on features. It can be considered as a set of if-then rules, which also can be thought of as conditional probability distributions defined in feature space and class space. Decision tree uses tree structure and the tree begins with a single node representing the training samples. If the samples are all in the same class, the node becomes the leaf and the class marks it. Otherwise, the algorithm chooses the discriminatory attribute as the current node of the decision tree. According to the value of the current decision node attribute, the training samples are divided into several subsets, each of which forms a branch, and there are several values that form several branches. For each subset or branch obtained in the previous step, the previous steps are repeated, recursively forming a decision tree on each of the partitioned samples.



**Figure 3.1.4(e) Decision Tree**

**vi) Naive Bayes :** Naive Bayes is a classification technique with a notion which defines all features as independent and unrelated to each other. It defines that the status of a specific feature in a class does not affect the status of another feature. Since it is based on conditional probability it is considered as a powerful algorithm employed for classification purposes. It works well for the data with imbalancing problems and missing values. Naive Bayes is a machine learning classifier which employs the Bayes Theorem. Using Bayes theorem posterior probability  $P(C|X)$  can be calculated from  $P(C)$ ,  $P(X)$  and

$P(X|C)$ . Therefore,  $P(C|X) = (P(X|C) P(C))/P(X)$  Where,  $P(C|X)$  = target class's posterior probability .  $P(X|C)$  = predictor class's probability.  $P(C)$  = class C's probability being true.  $P(X)$  = predictor's prior probability.



**Figure 3.1.4(f) Naïve Bayes**

### **3.1.5 Model Building :**

This is the most important phase which includes model building for prediction of diabetes. In this we have implemented various machine learning algorithms which are discussed above for diabetes prediction.

#### **Procedure of Proposed Methodology-**

##### **Step1:**

Import required libraries, Import diabetes dataset.

##### **Step2:**

Pre-process the data to remove all the null values and missing data.

##### **Step3:**

Perform a percentage split of 80% to divide the dataset as Training set and 20% to Test set.

**Step4:**

Select the machine learning algorithm i.e. KNearestNeighbour, Support Vector Machine, Decision Tree, Logistic regression, Random Forest, SVM, Naïve Bayes algorithm.

**Step5:**

Build the classifier model for the mentioned machine learning algorithm based on the training set.

**Step6:**

Test the Classifier model for the mentioned machine learning algorithm based on the test set.

**Step7:**

Perform Comparison Evaluation of the experimental performance results obtained for each classifier.

**Step8:**

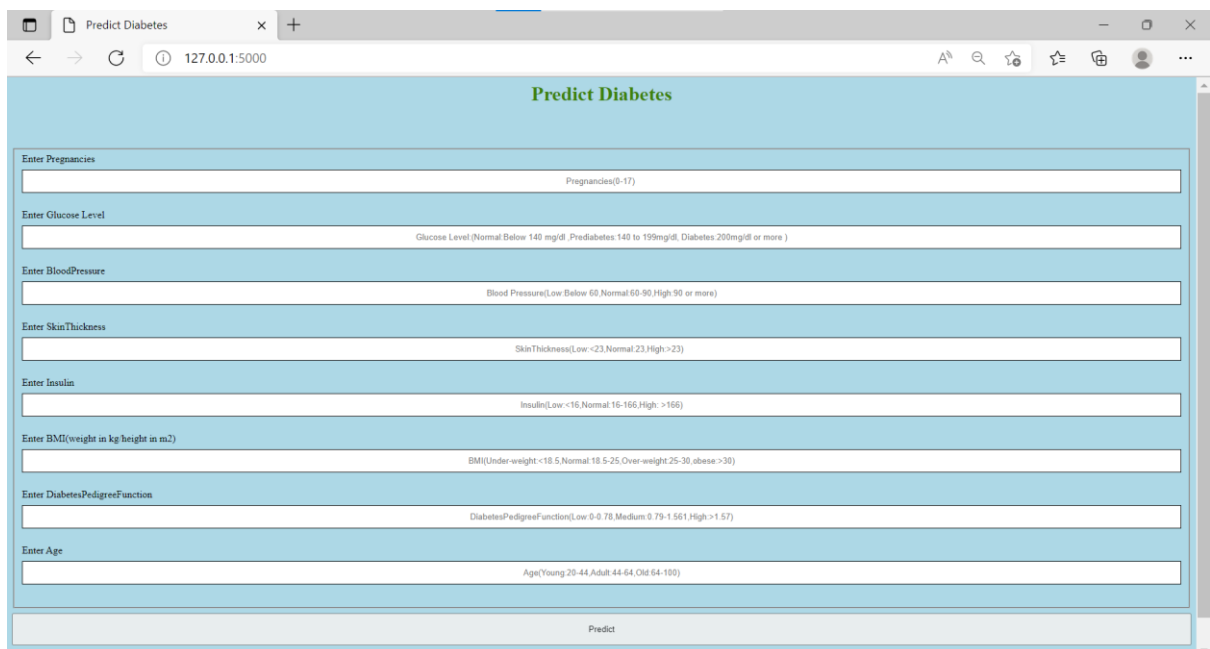
After analyzing based on various measures conclude the best performing algorithm.

**Step9:**

Finally build and Host a Flask Web app.

**Step10:**

A user has to put details like Pregnancies, Insulin Level, Age, BMI, DiabetesPedigree Function, Skin Thickness, Glucose Level etc



The screenshot shows a web browser window with the title 'Predict Diabetes'. The URL bar shows '127.0.0.1:5000'. The page has a light blue header with the title 'Predict Diabetes' in green. Below the header, there are several input fields with labels and placeholder text:

- Enter Pregnancies:** Pregnancies(0-17)
- Enter Glucose Level:** Glucose Level(Normal Below 140 mg/dl ,Prediabetes 140 to 199mg/dl, Diabetes 200mg/dl or more )
- Enter BloodPressure:** Blood Pressure(Low Below 60 ,Normal 60-90 ,High 90 or more)
- Enter SkinThickness:** SkinThickness(Low:<23 ,Normal 23 ,High>23)
- Enter Insulin:** Insulin(Low<16 ,Normal 16-166 ,High >166)
- Enter BMI(weight in kg,height in m2):** BMI(Under-weight <18.5 ,Normal 18.5-25 ,Over-weight 25-30 ,obese >30)
- Enter DiabetesPedigreeFunction:** DiabetesPedigreeFunction(low 0-0.78 ,Medium 0.79-1.561 ,High >1.57)
- Enter Age:** Age(Young 20-44 ,Adult 44-64 ,Old 64-100)

At the bottom of the form is a 'Predict' button.

**Figure 3.1.5 Web Page**

**Step11:**

Once it gets the Fields information, the result is displayed on new page

## 4. SIMULATION RESULTS AND ANALYSIS

In this work different steps were taken. The proposed approach uses different classification and ensemble methods and is implemented using python. These methods are standard Machine Learning methods used to obtain the best accuracy from data. In this work we see that random forest classifiers achieve better compared to others. Overall we have used the best Machine Learning techniques for prediction and to achieve high performance accuracy. Table shows the result of the Machine Learning methods.

ALGORITHM USED	ACCURACY
Random Forest	<b>97%</b>
Decision Tree	96%
Svm	81.25%
kNearestNeighbours	80.25%
Logistic Regression	78.75%
Naive Bayes	75.5%

**Table 4 Results of algorithms**

These are the results which we got after applying the machine learning algorithms

## 4.1 RESULTS

### Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train,y_train)
y_pred=gnb.predict(X_test)
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",gnb.score(X_train,y_train)*100)
print("Accuracy:",accuracy_score(y_test,y_pred)*100)
```

Classification Report is:

	precision	recall	f1-score	support
0	0.82	0.82	0.82	269
1	0.63	0.63	0.63	131
accuracy			0.76	400
macro avg	0.72	0.72	0.72	400
weighted avg	0.76	0.76	0.76	400

Confusion Matrix:

```
[[220  49]
 [ 49  82]]
```

Training Score:

75.5625

Accuracy: 75.5

## Logistic Regression

```
» #implementation of Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import classification_report
model = LogisticRegression()
model = model.fit(X_train, y_train)
predictions=model.predict(X_test)
print(classification_report(y_test,predictions))
print("confusion matrix:",confusion_matrix(y_test,predictions))
print("Training Score: ", model.score(X_train, y_train))
print("Testing Score: ", model.score(X_test, y_test))
print("Accuracy:",accuracy_score(y_test,predictions)*100)
```

	precision	recall	f1-score	support
0	0.81	0.89	0.85	269
1	0.72	0.58	0.64	131
accuracy			0.79	400
macro avg	0.76	0.73	0.75	400
weighted avg	0.78	0.79	0.78	400

confusion matrix: [[239 30]  
[ 55 76]]  
Training Score: 0.7725  
Testing Score: 0.7875  
Accuracy: 78.75

## KNearestNeighbors

```
: ▶ #implementation of KNearestNeighbors
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
classifier = KNeighborsClassifier(n_neighbors=8,p=2,metric='euclidean')
# fitting model
classifier.fit(X_train,y_train)
# making predictions
y_pred = classifier.predict(X_test)
# evaluating model
print("Classification Report is:\n",classification_report(y_test,y_pred))
conf_matrix = confusion_matrix(y_test,y_pred)
print("Confussion Matrix:",conf_matrix)
#print(f1_score(y_test,y_pred))
print("Training Score:\n",classifier.score(X_train,y_train)*100)
# accuracy

print("Accuracy:",accuracy_score(y_test,y_pred)*100)
```

Classification Report is:

	precision	recall	f1-score	support
0	0.81	0.92	0.86	269
1	0.78	0.56	0.65	131
accuracy			0.80	400
macro avg	0.79	0.74	0.76	400
weighted avg	0.80	0.80	0.79	400

Confussion Matrix: [[248 21]  
[ 58 73]]

Training Score:  
82.5

Accuracy: 80.25



## SVM

```
from sklearn.svm import SVC
svc = SVC(kernel='poly')
svc.fit(X_train, y_train)
y_pred = svc.predict(X_test)
print("classification Report is:\n", classification_report(y_test, y_pred))
print("confusion matrix", confusion_matrix(y_test, y_pred))
print("Training score:\n", svc.score(X_train, y_train)*100)
print("Accuracy:", accuracy_score(y_test, y_pred)*100)
```

Classification Report is:

	precision	recall	f1-score	support
0	0.81	0.94	0.87	269
1	0.81	0.56	0.66	131
accuracy			0.81	400
macro avg	0.81	0.75	0.77	400
weighted avg	0.81	0.81	0.80	400

Confusion Matrix:

```
[[252  17]
 [ 58  73]]
```

Training Score:

80.625

Accuracy: 81.25

## Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree.fit(X_train, y_train)
y_pred=dtree.predict(X_test)
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",dtree.score(X_train,y_train)*100)
print("Accuracy:",accuracy_score(y_test,y_pred)*100)
```

Classification Report is:

	precision	recall	f1-score	support
0	0.99	0.95	0.97	269
1	0.90	0.98	0.94	131
accuracy			0.96	400
macro avg	0.95	0.97	0.96	400
weighted avg	0.96	0.96	0.96	400

Confusion Matrix:  
[[255 14]  
[ 2 129]]  
Training Score:  
100.0  
Accuracy: 96.0

## Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
y_pred=rfc.predict(X_test)
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",rfc.score(X_train,y_train)*100)
print("Accuracy:",accuracy_score(y_test,y_pred)*100)
```

Classification Report is:

	precision	recall	f1-score	support
0	0.99	0.96	0.98	269
1	0.93	0.98	0.96	131
accuracy			0.97	400
macro avg	0.96	0.97	0.97	400
weighted avg	0.97	0.97	0.97	400

Confusion Matrix:  
[[259 10]  
[ 2 129]]  
Training Score:  
100.0  
Accuracy: 97.0

## Predict Diabetes through Web Application:

**Case1:** Outcome Negative(i.e A person not having Diabetes)

The image displays two browser windows. The top window, titled 'Predict Diabetes', shows a form with the following inputs: Pregnancies (1), Glucose Level (89), Blood Pressure (Diastolic) (66), Skin Thickness (23), Insulin (94), BMI (weight in kg height in m2) (21), Diabetes Pedigree Function (0.167), and Age (21). The bottom window, titled 'Diabetes Predictor', shows the same form with the 'Predict' button highlighted.

Input Field	Value
Enter Pregnancies	1
Enter Glucose Level	89
Enter BloodPressure(Diastolic)	66
Enter SkinThickness	23
Enter Insulin	94
Enter BMI(weight in kg height in m2)	21
Enter DiabetesPedigreeFunction	0.167
Enter Age	21
Predict	

**Chances of Getting Diabetes is Low**

**Figure 4.1(a) Result1**

**case2:**Outcome Positive(I.e, A person having Diabetes):

The image displays two browser windows. The top window, titled 'Predict Diabetes', shows a form with the following inputs: Pregnancies: 2, Glucose Level: 148, BloodPressure(Diastolic): 72, SkinThickness: 35, Insulin: 94, BMI(weight in kg height in m.): 26, DiabetesPedigreeFunction: 1.56, and Age: 41. The bottom window, titled 'Diabetes Predictor', shows the same form with a 'Predict' button at the bottom.

Field	Value
Enter Pregnancies	2
Enter Glucose Level	148
Enter BloodPressure(Diastolic)	72
Enter SkinThickness	35
Enter Insulin	94
Enter BMI(weight in kg height in m.)	26
Enter DiabetesPedigreeFunction	1.56
Enter Age	41

**Chances of having Diabetes is more**

**Figure 4.1(b) Result2**

## **5. CONCLUSION**

### **5.1 CONCLUSION**

The main aim of this project was to design and implement Diabetes Prediction Web Application Using Machine Learning Methods and Performance Analysis of those methods and it has been achieved successfully. The proposed approach uses various classification and ensemble learning methods in which SVM, KNearestNeighbours, Random Forest, Decision Tree, Logistic Regression and Naive Bayes are used. 97% classification accuracy has been achieved by Random Forest. The Experimental results can assist health care to make early predictions and make early decisions to cure diabetes and save humans life.

### **5.2 FUTURE SCOPE**

In future, the designed system with the used machine learning classification algorithms can be used to predict or diagnose other diseases. The work can be extended and improved for the automation of diabetes analysis including some other machine learning algorithms.

## REFERENCES

1. Perveen, Sajida; Shahbaz, Muhammad; Guergachi, Aziz; Keshavjee, Karim (2016). *Performance Analysis of Data Mining Classification Techniques to Predict Diabetes*. *Procedia Computer Science*, 82(), 115–121. doi:10.1016/j.procs.2016.04.016
2. Sisodia, Deepti; Sisodia, Dilip Singh (2018). *Prediction of Diabetes using Classification Algorithms*. *Procedia Computer Science*, 132(), 1578–1585. doi:10.1016/j.procs.2018.05.122
3. Yahyaoui, A., Jamil, A., Rasheed, J., & Yesiltepe, M. (2019). *A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques*. *2019 1st International Informatics and Software Engineering Conference (UBMYK)*. doi:10.1109/ubmyk48245.2019.89655
4. .Kandhasamy, J. Pradeep; Balamurali, S. (2015). *Performance Analysis of Classifier Models to Predict Diabetes Mellitus*. *Procedia Computer Science*, 47(), 45–51. doi:10.1016/j.procs.2015.03.182
5. Kannadasan, K; Edla, Damodar Reddy; Kuppili, Venkatanaresbhabu (2018). *Type 2 diabetes data classification using stacked autoencoders in deep neural networks*. *Clinical Epidemiology and Global Health*, (), S221339841830277X–. doi:10.1016/j.cegh.2018.12.004
6. Kumar, P. Suresh; Pranavi, S. (2017). *[IEEE 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS) - Dubai, United Arab Emirates (2017.12.18-2017.12.20)] 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS) - Performance analysis of machine learning algorithms on diabetes dataset using big data analytics.*, (), 508–513. doi:10.1109/ICTUS.2017.8286062
7. Al Jarullah, Asma A. (2011). *[IEEE 2011 International Conference on Innovations in Information Technology (IIT) - Abu Dhabi, United Arab Emirates (2011.04.25-2011.04.27)] 2011 International Conference on Innovations in Information Technology - Decision tree discovery for the diagnosis of type II diabetes.*, (0), 303–307. doi:10.1109/innovations.2011.5893838
8. B.M. Patil; R.C. Joshi; Durga Toshniwal (2010). *Hybrid prediction model for Type-2 diabetic patients.*, 37(12), 8102–8108. doi:10.1016/j.eswa.2010.05.078

9. Zhu, Changsheng; Idemudia, Christian Uwa; Feng, Wenfang (2019). *Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. Informatics in Medicine Unlocked*, (), 100179–. doi:10.1016/j.imu.2019.100179
  
10. Lakshmi, V. Swathi; Nithya, V.; Sripriya, K.; Preethi, C.; Logeshwari, K. (2019). *[IEEE 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN) - Pondicherry, India (2019.3.29-2019.3.30)] 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN) - Prediction of Diabetes Patient Stage Using Ontology Based Machine Learning System.* , (), 1–4. doi:10.1109/ICSCAN.2019.8878831